Лабораторная работа 9

Разработка Web-приложений в Delphi

Цель работы: Получить навыки использования Web-ориентированных компонент Delphi при разработке приложений.

Ход выполнения работы

Задание 1. Средствами Delphi создать простую Web-страницу.

1. Запустите IDE Delphi и выполните команду: **File – New - Other**

Из представленных в новом окне компонентов выберите Text и кликните на кнопке OK.

2. В новой вкладке File1.txt наберите код HTML-страницы:

```
<HTML>
<HEAD>
<TITLE>Это заголовок моей страницы</TITLE>
</HEAD>
<BODY BGCOLOR="#F0F8FF">
<H2><i><FONT COLOR="#8B0000"> В Delphi можно создавать сайты
</i></H2>
<BR>
<h3 ALIGN=CENTER><FONT COLOR="#000000">
Как это сделать?</h3>
<br>
Довольно просто. Достаточно создать текстовый документ и
сохранить его в файл с расширением  HTML.
<br>
Затем открыть этот файл в браузере.
<br>
<b><i>Примечание</i></b>:
<br>
Главное надо знать,
<br>
<P align="center"> <FONT size="12" color="#ff0000"> ЧТО
ПИСАТЬ В ЭТОМ ФАЙЛЕ! </FONT></P>
<br><br><br>
Список советов доступен по этой
<a href="http://www.wisdomweb.ru/HTML5/"> ссылке </a>
</BODY>
</HTML>
```

3.Выполните команду File – Save as… и сохраните код HTML-страницы в файл под именем Index.html

Для просмотра страницы выберите вкладку Preview внизу редактора кода.

Для возврата к коду страницы выберите вкладку Code.

4.Простая Web-страница готова, ее можно открыть для просмотра при помощи любого браузера.

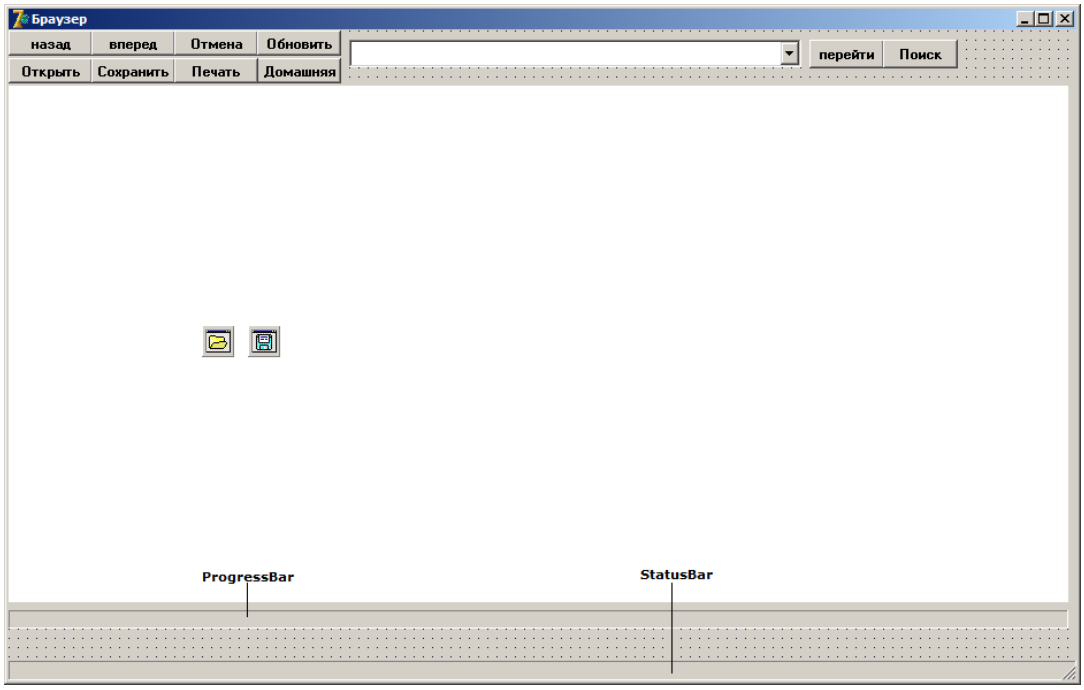
Задание 2. Разработать программу для просмотра Web-страниц.

1. Создайте новый проект.

Разместите на форме компоненты:

- 1 компонент WebBrowser со вкладки Internet ,
- 10 компонентов Button,
- 1 компонент ComboBox,
- 1 - компонент ProgressBar,
- 1 – компонент StatusBar,
- 1 - компонент OpenDialog
- 1 - компонент SaveDialog.

Разместите их на форме примерно следующим образом:



В **uses** подключаем **MSHTML и ActiveX**

2. Создадим событие **OnCreate** для формы и запишем в нем следующий код:

```
WebBrowser1.Navigate('http://www.google.com');
```

Вместо google.com можно ввести адрес любой странички, которую бы вы хотели видеть при запуске браузера.

Если нет доступа к сети Интернет, то для отображения разработанной в задании 1 страницы в обработчик события OnCreate для формы запишите код

```
    WebBrowser1.Navigate(ExtractFilePath(ParamStr(0)) +
'Index.html')
```

3. Создадим событие OnClick на Перейти и в коде напишем:

```
    var
    a:string;
    begin
    WebBrowser1.Navigate(ComboBox1.Text);  // Передаем в поиск,
введенный адрес из ComboBox1;
    a:=ComboBox1.text;  // Записываем параллельно этот адрес в
переменную;
```

```
    ComboBox1.Items.Add(a);  // Загружаем введенное в
список  ComboBox1;
    end;
```

Процедура будет позволять нам при нажатии на ComboBox просматривать историю вводимых вручную адресов, в строку браузера.

Далее для удобства сделаем так, чтобы данный механизм работал не только от кнопки Перейти, но и от нажатия нами на клавишу Enter, для этого:

Создадим событие **OnKeyPress** у компонента **ComboBox1** запишем следующее:

```
    var
    a:string;
    begin
    if Key=#13 then
    begin
    WebBrowser1.Navigate(ComboBox1.Text);
    a:=ComboBox1.text;
    ComboBox1.Items.Add(a);
    end;
    end;
```

4. Для кнопки **Назад,** пишем в событии **OnClick** :

```
    var
    Doc, Histor: Variant;
    begin
    Doc:= WebBrowser1.OleObject.document;  {Ссылаемся на
интерфейс документа}
    if TVarData(Doc).VDispatch = nil then Exit;  {Если документ
отсутствует, тогда выходим}
    Histor:=Doc.parentWindow.history;
    Histor.go(-1);
    end;
```

5. Для кнопки - **Вперед** пишем в событии **OnClick** :

```
    var
    Doc, Histor: Variant;
    begin
    Doc:= WebBrowser1.OleObject.document;
    if TVarData(Doc).VDispatch = nil then Exit;
    Histor:=Doc.parentWindow.history;
    Histor.go(1);
    end;
```

6. Для перехода на домашнюю страничку в **OnClick** кнопки **Домашняя** впишите

```
    WebBrowser1.GoHome;
```

7. Для остановки загрузки в **OnClick** на **Отмена** пишем**;**

```
    WebBrowser1.Stop;
```

8. Для обновления странички в **OnClick** на **Обновить** пишем;

```
WebBrowser1.Refresh;
```

9. Для открытия файлов в нашем браузере расположенных на локальных дисках в **OnClick** на **Открыть** пишем:

```
if OpenDialog1.Execute then
begin
WebBrowser1.Navigate(OpenDialog1.FileName);
ComboBox1.Text:=OpenDialog1.FileName;
end;
```

10. Для возможности нашего браузера также сохранять странички на компьютер, в **OnClick** на **Сохранить** пишем:

```
WebBrowser1.ExecWB(OLECMDID_SAVEAS,
OLECMDEXECOPT_DODEFAULT);  //Сохраняем содержимое WebBrowser1;
```

11. Для получения возможности печати по нажатию на кнопке **Печать в OnClick** для этой кнопки пропишем:

```
var
PostData, Headers: OLEvariant;
begin
WebBrowser1.ExecWB(OLECMDID_PRINT, OLECMDEXECOPT_DODEFAULT,
PostData, Headers);  //Печатаем содержимое TWebBrowser;
end;
```

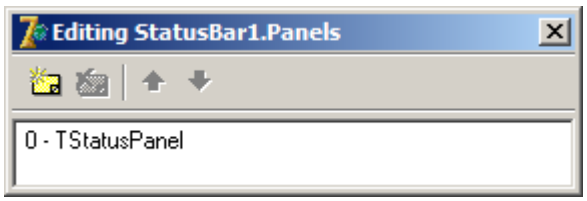
12. Теперь организуем поиск слов на страничке, и в **OnClick** на **Поиск** пишем:

```
begin
WebBrowser1.ExecWB(OLECMDID_FIND,
OLECMDEXECOPT_DODEFAULT);  //Поиск слов на сайте;
end;
```

13. Чтобы **ProgressBar1** отображал ход загрузки страниц, в событии **OnProgressChange** компонента **WebBrowser1**, нужно прописать:

```
procedure TForm1.WebBrowser1ProgressChange(Sender: TObject;
Progress,
  ProgressMax: Integer);
begin
ProgressBar1.Max:=ProgressMax;
ProgressBar1.Position:=Progress;
end;
```

14. Для отображения **URL** хода загрузки выделите компонент **StatusBar1 и активизируйте его свойство Panels. Используя пункт Add контекстного меню, создайте для компонента панель**



В событии **OnStatusTextChange** компонента **WebBrowser1**, пропишите:

```
StatusBar1.Panels[0].text:=text;
```

15. Сохраните проект и проверьте его работу.

Контрольные вопросы

1. Компонент WebBrowser.
2. Метод navigate компонента WebBrowser.